COMPARISON OF UNSUPERVISED ANOMALY DETECTION METHODS FOR SYSTEMS HEALTH MANAGEMENT USING SPACE SHUTTLE MAIN ENGINE DATA*

R. A. Martin, M. Schwabacher, N. Oza, and A. Srivastava
Intelligent Systems Division
NASA Ames Research Center
Moffett Field, CA

ABSTRACT

Several different unsupervised anomaly detection algorithms have been applied to Space Shuttle Main Engine (SSME) data to serve the purpose of developing a comprehensive suite of Integrated Systems Health Management (ISHM) tools. As the theoretical bases for these methods vary considerably, it is reasonable to conjecture that the resulting anomalies detected by them may differ quite significantly as well. As such, it would be useful to apply a common metric with which to compare the results. However, for such a quantitative analysis to be statistically significant, a sufficient number of examples of both nominally categorized and anomalous data must be available.

Due to the lack of sufficient examples of anomalous data, use of any statistics that rely upon a statistically significant sample of anomalous data is infeasible. Therefore, the main focus of this paper will be to compare actual examples of anomalies detected by the algorithms via the sensors in which they appear, as well the times at which they appear. We find that there is enough overlap in detection of the anomalies among all of the different algorithms tested in order for them to corroborate the severity of these anomalies. In certain cases, the severity of these anomalies is supported by their categorization as failures by experts, with realistic physical explanations. For those anomalies that can not be corroborated by at least one other method, this overlap says less about the severity of the anomaly, and more about their technical nuances, which will also be discussed.

INTRODUCTION

A comprehensive suite of failure detection algorithms can be used to aid in the early detection of spacecraft propulsion engine anomalies and potential failures during operation. The study provided in this paper reviews algorithms that have been applied to SSME data as a testbed platform, in anticipation of applying them to future spacecraft propulsion systems such as the Ares I crew launch vehicle, and Ares V, the heavy lift cargo launch vehicle. It is well known by algorithm designers that having more than a single means to detect failure aids in corroboration and also builds in redundancy. Several architectures have been developed that support this very concept, with one prime example by Park et. al.[1] in which SSME data was also used as the target for building the model. As such, it is very possible that a preliminary architecture consisting of the algorithms applied to the SSME data presented in this study may be developed to support engine anomaly detection for future spacecraft propulsion systems.

The methods that we will study in the paper all have a similar modeling paradigm in common. All models are data–driven and unsupervised, meaning that a nominal representation is generated purely from the data, without any portion of the model that integrates rules, component level or first principles physics-based representation of the subject platform. From a statistical standpoint, due to limited examples of failures, the supervised or semi-supervised paradigm for machine learning will not produce

Distribution Statement A: Approved for public release; distribution is unlimited

*This effort was supported by the Liquid Propulsion portion of NASA’s Integrated System Health Management project within the Exploration Technology Development Program funded by the Exploration Systems Mission Directorate.

an acceptable model. Both of these methods require a modest number of labeled examples of failures of anomalies. As such, we apply the unsupervised paradigm for all of the methods to be used, in which models are trained on data that represent nominal operation only. In each of the following subsections, brief descriptions and relevant references are provided for each of the methods to be applied.

ORCA

Orca is a software tool that uses a nearest neighbor based approach to outlier detection which may be based upon the Euclidean distance metric. It uses a modified pruning rule that allows for increased computational efficiency, running in near linear time. More information on this algorithm and some of its applications can be found in Bay and Schwabacher[2], and Schwabacher[3]. This algorithm outputs a total score which represents the aggregation of scores for all contributing factors, as well as the average distance to the nearest *k* neighbors.

IMS (INDUCTIVE MONITORING SYSTEM)

IMS, the Inductive Monitoring System, is a software tool that performs outlier detection by learning the bounds of clusters in multi-dimensional feature space for nominal operation during the model training phase. During the monitoring phase, any points falling outside of hypercubes defined and stored in a system behavior cluster database that represent the bounds of nominal operation are considered outliers. This algorithm outputs a score that represents the Euclidean distance between the monitored point and the nearest cluster. More information on this algorithm can be found in Iverson[4].

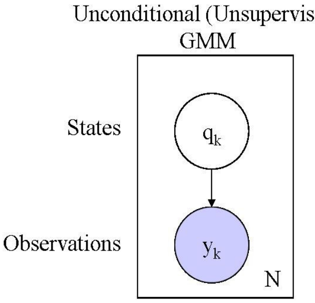
GRITBOT

GritBot is a commercially available data mining software tool that performs anomaly detection using a decision tree-based approach. The resulting rules for categorization of potential anomalies are presented individually, at particular times. As such, there is not an overall score provided, or even a parameter-based score for an entire time series. Rather, anomalies are listed in ranked order according to their statistical significance. Each identified anomaly presents the time at which the anomaly appears, the parameter-based decision rule listed with corresponding relevant statistics, and the respective parameter values of the identified anomaly.

GMM (GAUSSIAN MIXTURE MODEL)

The Gaussian mixture model is derived from Bayesian statistics in the sense that it can be easily represented within the probabilistic graphical modeling paradigm (also sometimes referred to as Bayes' nets). An example of a graphical model representing the Gaussian mixture model is presented in Fig. 1.

Figure 1: Graphical Model Representation of GMM



$$q_k \in \{1, \ldots, M\}$$

$$\theta = (\alpha_1, \ldots, \alpha_M, \mu_1, \ldots, \mu_M, \Sigma_1, \ldots, \Sigma_M)$$

The shaded nodes represent the observed continuous-valued data, y_k at the time instant k. The unshaded nodes, q_k, represent M unobserved discrete variables that are conditioned on the observed data in order to form sufficient statistics. These sufficient statistics in turn are used to generate estimates of the parameters, θ, for each of the M Gaussian mixtures, including mixture weights (α_i), means (μ_i) and covariance matrices (Σ_i). An iterative learning process is implemented via the EM algorithm to converge at values for these estimates. One important assumption is implicitly made about the data using this modeling paradigm, which is that all N observations are temporally independent. This same assumption is also tacitly implied in the previous methods discussed.

There are several variants of the Gaussian mixture model that can be investigated in this particular study. They include variants based on the choice of correlation among parameters via constraints on the covariance matrix for a multivariate GMM in which all sensors form a single feature vector. Other variants include the application of various data reduction techniques for which a univariate GMM is trained, or a univariate GMM which is trained on each individual sensor (parameter). In the latter case, either a single alarm system for all sensors or multiple alarm systems for each sensor can be designed for these univariate GMM's. Training multiple alarm systems for individual univariate GMM's per parameter value denies the ability to take advantage of correlations among different sensors, however, it inherently allows for the ability of anomalies to be isolated or localized. The design of alarm systems is performed by selecting a threshold based upon the log-likelihood function value of the distribution. An alarm is triggered upon evaluation of the log-likelihood value at y_k exceeding this predefined threshold. The negative log-likelihood value also serves as a scoring metric for all time instants. A previous study[5] provides a more thorough discussion on the theory of designing the alarm systems and model development in detail. Exploration of the all of the variants discussed here is also provided in [5], using SSME data as the basis of the study. As such, for comparative purposes we will investigate only the variant that provides the ability to localize anomalies to a particular sensor in this study.

LDS (LINEAR DYNAMIC SYSTEM)

The Linear Dynamic System has roots in various research communities, including machine learning and control theory. Unlike the previous general purpose anomaly detection methods, we attempt to address a very application-specific anomaly detection problem by appealing to the use of the Linear Dynamic System from both research perspectives. Specifically, we use the SSME throttle control system error (Fig. 2) as the basis of observations to train a model that can also be expressed in the probabilistic graphical modeling framework (Fig. 3), a DBN (Dynamic Bayes' network) in this case.

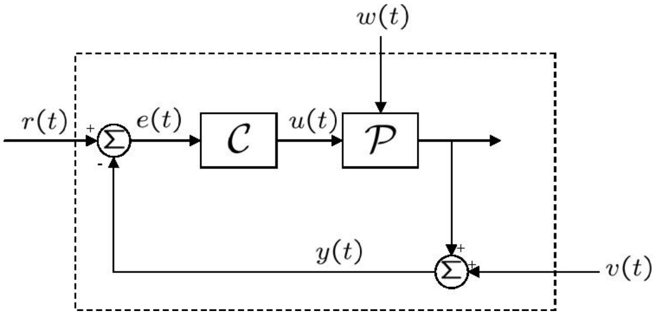


Figure 2: Throttle Control System Error…e(t)

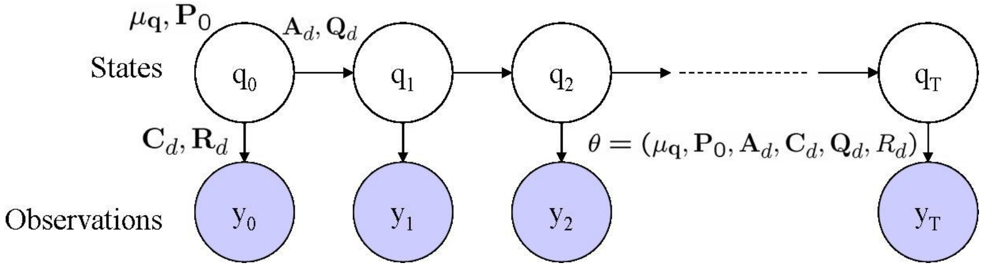

$\mu_{\mathbf{q}}, \mathbf{P}_0$
$\mathbf{A}_d, \mathbf{Q}_d$
States
q_0
q_1
q_2
q_T
$\mathbf{C}_d, \mathbf{R}_d$
$\theta = (\mu_{\mathbf{q}}, \mathbf{P}_0, \mathbf{A}_d, \mathbf{C}_d, \mathbf{Q}_d, R_d)$
Observations
y_0
y_1
y_2
y_T

Figure 3: DBN Representation of a Linear Dynamic System

Also unlike the previous cases, the observations are serially correlated, and 2nd order dynamics are assumed. However, the model training and alarm system design procedure are identical to the details provided for the GMM. As such, again the negative log-likelihood value may serve as a scoring metric for all time instants. The study cited previously[5] also uses this modeling paradigm as a case for study. More thorough discussion on the details of blending both machine learning and control theory to serve the purpose of anomaly detection is provided there.

ONE-CLASS SUPPORT VECTOR MACHINE

The one-class support vector machine is a very specific instance of a support vector machine which is geared for anomaly detection. The generic support vector machine (SVM) can be used to classify data in multiple dimensions by finding an appropriate decision boundary. Unlike neural networks, the support vector machine finds the boundaries that provide the maximum margin between different classes of data. Additionally, using the support vector machine one can map data from a lower dimensional space that is not linearly separable to a higher (even infinite-dimensional) space where the data are linearly separable by a hyperplane. This is performed by using what is commonly known in machine learning as the “kernel trick,” when using SVM’s. A kernel function is chosen to map the data from the lower-dimensional space to the higher dimensional space. It can be chosen arbitrarily so as to best suit the data and at the same time reduce the computational burden involved with generating the mapped values by direct evaluation. “Support vectors” correspond to those points that lie along the margin or closest to it. The maximum margin between classes is found by solving a quadratic optimization problem.

The one-class SVM differs from the generic version of the SVM in that the quadratic optimization problem includes an allowance for a certain small predefined percentage of outliers, making it suitable for anomaly detection. These outliers lie between the origin and the optimal separating hyperplane. All the remaining data fall on the opposite side of the optimal separating hyperplane, belonging to a single, nominal class, hence the terminology “one-class” SVM. The SVM outputs a score that represents the distance from the data point being tested to the optimal hyperplane. Positive values for the one-class SVM output represent normal behavior (with higher values representing greater normality) and negative values represent abnormal behavior (with lower values representing greater abnormality). More technical details on the one-class SVM are available in Das et al. [6] and Cohen et al.[7].

RESULTS AND DISCUSSION

Recall the purpose of this paper, which is to perform a comparative analysis among the different methods applied to the same SSME dataset. Table 1 illustrates the data used for both training and validation. Portions of the data representing startup, shutdown, and major throttling transients have been eliminated in order to prevent spurious false alarms that all algorithms are susceptible to during validation.

Data Sources	Training	Validation	
	Nominal	Nominal	Potential Anomalies
Flight Data	STS-77 (#1)	STS-103 (#2)	STS-77 (#2)
	STS-78 (#1)	STS-103 (#3)	STS-91 (#1)
	STS-78 (#2)	STS-106 (#1)	STS-93 (#1)
	STS-78 (#3)	STS-106 (#2)	STS-93 (#3)
Test Stand Data	A10851	A10852	A10853
	A20726	A20750	A20619

Table 1: SSME training and validation datasets

In adherence with the unsupervised machine learning paradigm, all of the six training data files are categorized as nominal. Both flight data and test stand data are represented, adding to the richness and heterogeneity of the resulting models built upon this dataset, in part due to the variety of operational conditions experienced throughout both flight and testing runs. The validation data is split into two sets of six, one set that represents nominal data, and the other that represents potentially anomalous data. The nominally categorized data has for the most part been found or thought to be free of any significant anomalies. However, in some instances benign anomalies may appear in the validation of nominally categorized data where there was no prior suspicion of them. The potentially anomalous data has been categorized as such by experts in the case of the test stand data, where in the other in-flight cases there have been documented failures or anomalies found by other algorithms in independent analyses not discussed here, i.e. Bickford[8]. A subjective evaluation for the relative severity of the potentially anomalous data is provided in Table 2 below.

Failure Data	Failure Type	Functional Categorization	Severity
STS-77 (#2)	Anomalous Spike in Sensor Reading	Controller	Mild
STS-91 (#1)	Sensor Failure	Controller	Mild
STS-93 (#1)	Controller Failure	Controller	Moderate
STS-93 (#3)	Fuel Leak and Controller Failure	Controller	Moderate to Severe
A20619	Knife Edge Seal Crack	Vibration	Moderate to Severe
A10853	Turbine Blade Failure	Vibration	Severe

Table 2: Characterization of Failures

The functional categorization is meant to provide context for the type of sensor that the failure presents in. Controller data includes sensor measurements such as valve positions, pressures, temperatures, etc. Vibration data are primarily accelerometer measurements used to assess the structural integrity of the engine. We now present the results for all algorithms for validation examples shown in Table 1, including the potential anomalies shown in Table 2. Recall that the aim is to compare actual examples of anomalies detected by all algorithms presented. Of these techniques, only Orca, GritBot, and the GMM method will provide information pertaining to the sensors in which the potential anomalies appear. All methods provide the times at which the anomalies appear, conditioned on the fact that a specific threshold is used for the scoring metric. We can also determine if there is enough overlap in detection of the anomalies among all of the different algorithms tested in order for them to corroborate the severity of these anomalies as presented in Table 2.

Binary classification of the validation data will be performed by finding the threshold that splits the anomalous and nominally categorized files evenly. Our rationale for using this rule is due to there being exactly six examples of both nominal and anomalous validation data runs. With the exception of the GMM method and GritBot, at least one threshold exceedance by the score during the run is required to classify a file as anomalous. For the GMM method, at least one threshold exceedance by the score of an offending sensor is required during the run is required to classify a file as anomalous. As for GritBot, recall that there is no scoring metric, therefore, only the times of the anomalies will be shown, and the sensor in which they present. GritBot lists its anomalies in ranked order, and as such the top anomalies corresponding to the first six unique tests were chosen for display.

NOMINAL VALIDATION TRIALS

Our first example is nominally categorized SSME engine #2 for flight STS-103, with the results shown in Figure 4.

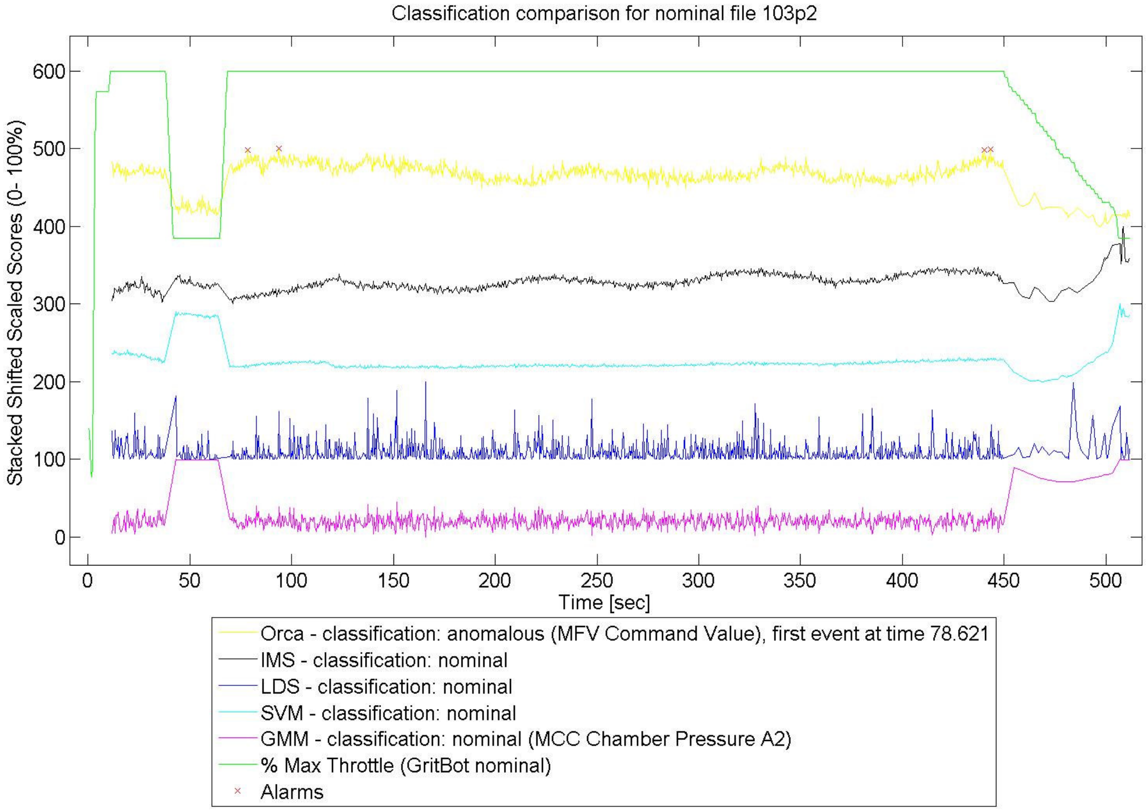


Figure 4: Anomaly Detection Comparison for Validation file STS-103, Engine #2

Figure 4 illustrates the scored values for all anomaly detection algorithms. Absolute scores are not provided, to allow for ease of display. Rather, the scores are all scaled to 100% based upon the maximum score encountered throughout the test, and shifted/stacked accordingly. The green line represents the throttle command, which is an apparent driver for many of the controller-based sensor readings that the SSME is outfitted with. As such, the score is often biased by the throttle command as well. This is evident for the scores shown in Figure 4. The only algorithm that classifies this test as anomalous is Orca, which has its first anomalous detection event around 79 seconds into the test. No significant operational event occurs at this time, and as such we can conclude that this is a false positive on Orca's part, as none of the remaining algorithms can corroborate an anomaly.

The second example is illustrated in Figure 5, which represents a nominally categorized SSME engine, #3, for the same flight a shown in Figure 4, STS-103.

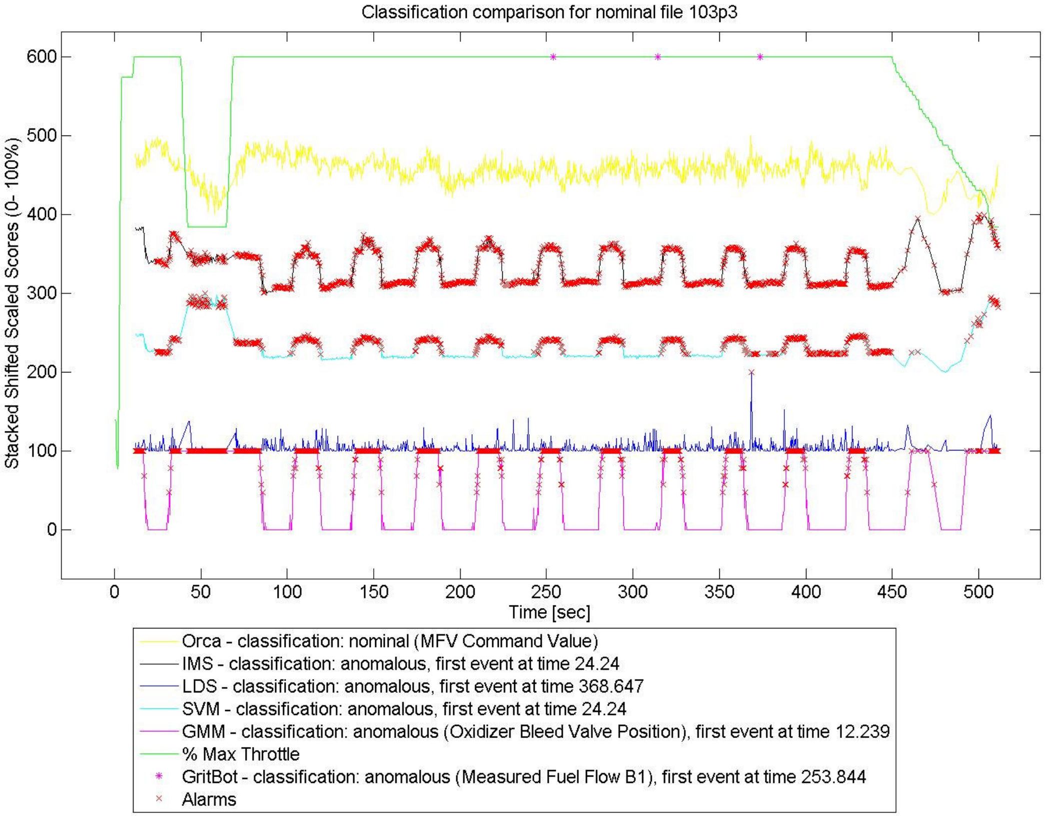


Figure 5: Anomaly Detection Comparison for Validation file STS-103, Engine #3

Here we see that the throttle profile is identical to the previous case, as expected since it is for the same flight. However, the scores for this engine are quite different. All algorithms with the exception of Orca classify this run as anomalous. Although the run was classified as nominal, it was found in a previous study[8] that this particular run experienced what was termed a "max noise failure." These failures were reported at times 38.1 sec and 72.74 sec for two high pressure fuel turbine discharge temperature sensors, respectively. The failure occurs due to rapid oscillations or repeated unexplained fluctuations. However, these particular sensors were not included as part of our analysis, and as such accounted for the nominal categorization of STS-103 #3. It is possible that the max noise failure was picked up in other sensors as well that manifested themselves in the apparent fluctuation shown in the scores in Fig. 5. This can be illustrated by the measured fuel flow sensor that the GMM method has isolated as anomalous. It is also noteworthy that the GMM method is the first to identify the max noise failure. Orca again is the one algorithm that votes against all others, this time as a false negative. The MFV Command Value shown on the legend for Orca is chosen as the largest contributing factor to the score.

The third nominal example is a different flight, STS-106, engine #1. In this case, GritBot is the only algorithm that breaks ranks with the others, finding an anomaly in the hydraulic system at 368.22 into the flight. This may be a false positive, as all other algorithms correctly categorize this as nominal run.

The fourth nominal example is for the same flight, STS-106, but for a different engine, #2, and the scores are shown in Fig. 6 below. Although this a nominally classified run, three of six algorithms pick out anomalous behavior, making a majority voting logic scheme inconclusive. For the GMM method, the selected threshold results in all points in time being picked out as anomalous for the MFV actuator position. IMS and LDS also show several times where the thresholds are exceeded by the alarm indicator, albeit to a lesser extent than for the GMM method. It may be possible that if the selected thresholds were incrementally higher, IMS and LDS would have yielded nominal categorizations for this run, and GMM would have provided the only anomalous classification. This would make the GMM method's classification a false positive, since the other algorithmic techniques don't corroborate this anomaly. However, due to the nature of the severity of the anomaly for the particular sensor identified by the GMM method, it may be worth further investigation.

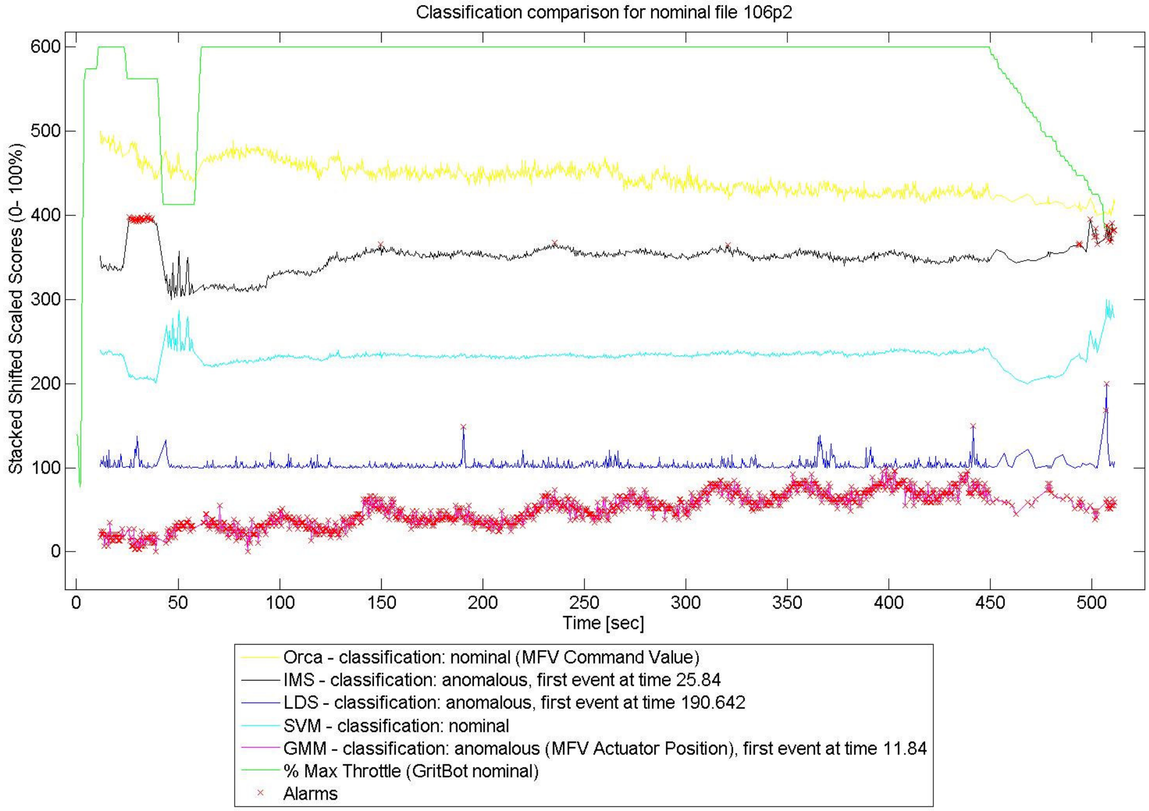


Figure 6: Anomaly Detection Comparison for Validation file STS-106, Engine #2

The final two nominal validation trials are based upon test stand data, the first of which is A10852. As illustrated in Figure 7, all algorithms applied to this test with the exception of Orca and GritBot classify this test as anomalous. The area of interest occurs from approximately 210 to 240 sec. For this particular test, during a period of nominal steady-state operation, a significant excursion in the scores of the remaining algorithms jump considerably. As it turns out, a planned mixture ratio change that deviates from what is normally encountered was executed precisely during these time periods. This is further substantiated by the fact the GMM method indicates this as the only sensor in which a significant anomaly appears. As such, it certainly stands to reason that this operational idiosyncrasy would present as an anomaly in remaining algorithms even though it was originally categorized as nominal. This anomaly also appeared in analysis for a completely independent study[9] that used an entropy-based method. With the exception of Orca and GritBot that provide a false negative report, all algorithms do report this test as anomalous. It is possible that the threshold was set too high for these methods to pick up the anomaly. It is also noteworthy that both the GMM and LDS methods identify the anomaly first, with the LDS method resulting in the earlier report of the anomaly. The final nominal test is for A20750, for which all algorithms with the exception of the GMM method categorize it as nominal. In this case the GMM method reports the anomaly to appear in the MFV command value sensor. This report is clearly a false positive, and may be due to the threshold value being set too low for this method.

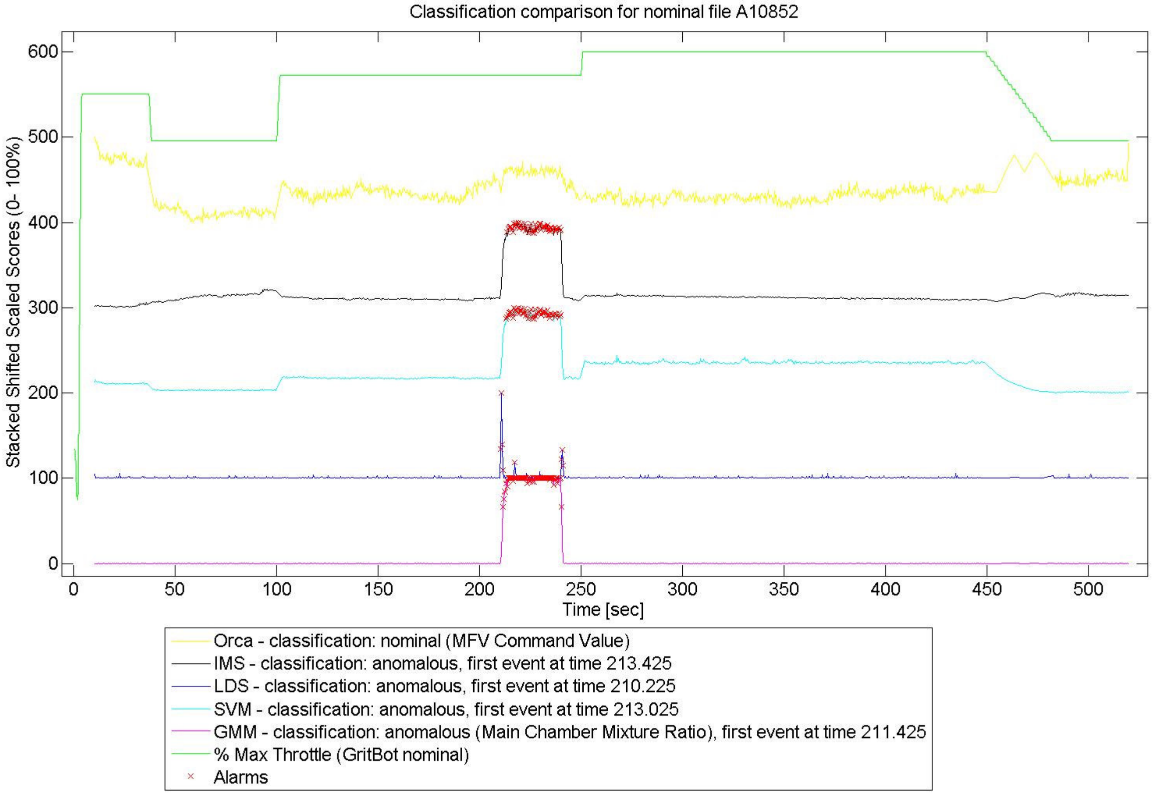


Figure 7: Anomaly Detection Comparison for Validation file A10852

ANOMALOUS VALIDATION TRIALS

Of the tests whose ground truth classification was anomalous, STS-77 engine #2 experienced an anomaly which was subjectively categorized to have the least severity, with it being an anomalous spike identified by analysis in previously cited study[8]. It was identified in a high pressure fuel pump discharge pressure sensor at 74.42 seconds. However, all of the algorithms categorize the validation trial as nominal. This result may be due to the fact that the thresholds for all methods were set too high in order to pick out this very mild anomaly that had not been validated by experts. The next anomalous test is listed in Table 2, for flight STS-91 engine #1, and represents a sensor failure that can also subjectively be categorized as mild or benign; however its severity is of greater importance than the anomalous spike from the previous example. In this case, Orca is the only algorithm that correctly categorizes the run as anomalous, even though the first detection occurs at 77.8 sec, and the largest contributing factor is the MFV command value. Ground truth indicates that the sensor failure occurred at 32.76 seconds, for the main combustion chamber pressure sensor.

The next two failures listed in Table 2 occurred on the same flight, STS-93, for both engines #1 and #3. Engine #1 experienced a controller failure due to an electrical spike on the main bus that resulted in a power transient onboard the orbiter. Engine #3 experienced the same, in addition to a hydrogen fuel leak caused by a ruptured cooling line. These details are publicly available on-line,[11] with additional supporting evidence provided by Bickford[8]. The controller failure is subjectively categorized to have only moderate severity due to the fact that back up controllers were automatically put into service. However, the fuel leak is potentially a major threat to engine operation, and as such is categorized as severe. For Engine #1, all algorithms with the exception of the LDS method recognized the failure using the chosen thresholds, and categorized it as anomalous. The first indication of the failure was reported by the GMM method at time 12.64 sec, almost 3 sec prior to GritBot's first indication of an anomaly, and over 12 sec prior to the remaining algorithms. The reason why the LDS incorrectly labeled this as nominal trial (i.e., yielded a false negative) is due to the fact that the signal being monitored by this method is the control system error. Since the failure was a total controller failure, both the commanded and actual sensor values zeroed out. As such, the control system error remained at zero, and resulted in no apparent evidence of an anomaly.

For the more severe failure in engine #3, all six algorithms correctly identified the anomaly, the first of which was the GMM method at 11.84 sec, followed by the LDS method at 17.44 sec. Both the GMM method and Orca isolated the main combustion chamber discharge temperature as a sensor that provides a significant contributing factor to the anomaly. Orca first identified the anomaly at 23.44 sec, as did IMS and the SVM method. However GritBot did not identify the anomaly until 176 sec, and it is a for a different sensor, the fuel preburner chamber pressure, followed by two other anomalies identified at 299 sec and 311 sec identified for the hydraulic system pressure sensor.

The final two examples of failures come from test stand trials. The first, test A20619, represents a material failure due to a knife edge seal crack in the high pressure oxidizer turbo pump. Without high frequency data and an algorithmic technique suited to analysis in the frequency domain, this type of failure may be overlooked by straightforward time series analysis on data that has a rather modest sampling rate. However, as shown in Fig. 8, we see that four out of six of the algorithmic techniques applied yield an anomalous classification. At least two of these classifications (particularly the LDS and SVM methods) are unrelated to the knife edge seal crack, and more likely related to the transient in the throttle that occurs just after 150 sec. Of course, we only train and validate for periods of steady-state operation, however, it is possible for some transients to have a more lasting effect than others. Orca and GritBot detect anomalies prior to this transient, localizing the MFV command value and the CCV actuator position sensors as the main contributing factors. These sensors do not necessarily have a direct bearing on the knife edge seal crack. Therefore, even though four sensors positively and correctly identified this test as anomalous, we cannot conclusively recommend a correct or false positive rating.

Finally, there has been a well documented[10] and researched failure for test A10853. For this test, a high pressure turbo pump blade failure occurred, and its onset can be readily identified via the composite high pressure fuel pump accelerometer for frequencies n the 50 – 800 Hz range, located at the 22.5 degrees radially. Figure 9 illustrates the votes of all algorithms applied to the test data for this run. As shown, all algorithms with the exception of the GMM method correctly classify it as anomalous. In this case, the threshold set for GMM method is too high, and as such yields a false negative. The offending sensor is known (the composite high pressure fuel pump accelerometer), so for illustrative purposes the GMM score is shown for this sensor. Orca is the first algorithm to detect an anomalous condition, well in advance of the actual onset of the failure at 20.66 sec, followed by the LDS method which identifies an anomalous condition just fractions of a second after the onset of the failure. Both GritBot and Orca pick out the measured fuel flow and the MFV command value sensors as the main contributing factors. GritBot also identifies additional sensors in which anomalies occur, which are the hydraulic system and controller internal pressures. However, none of these represent accelerometer measurements, or specifically the offending composite HPFP accelerometer; therefore isolation of the anomalies to these sensors is dubious.

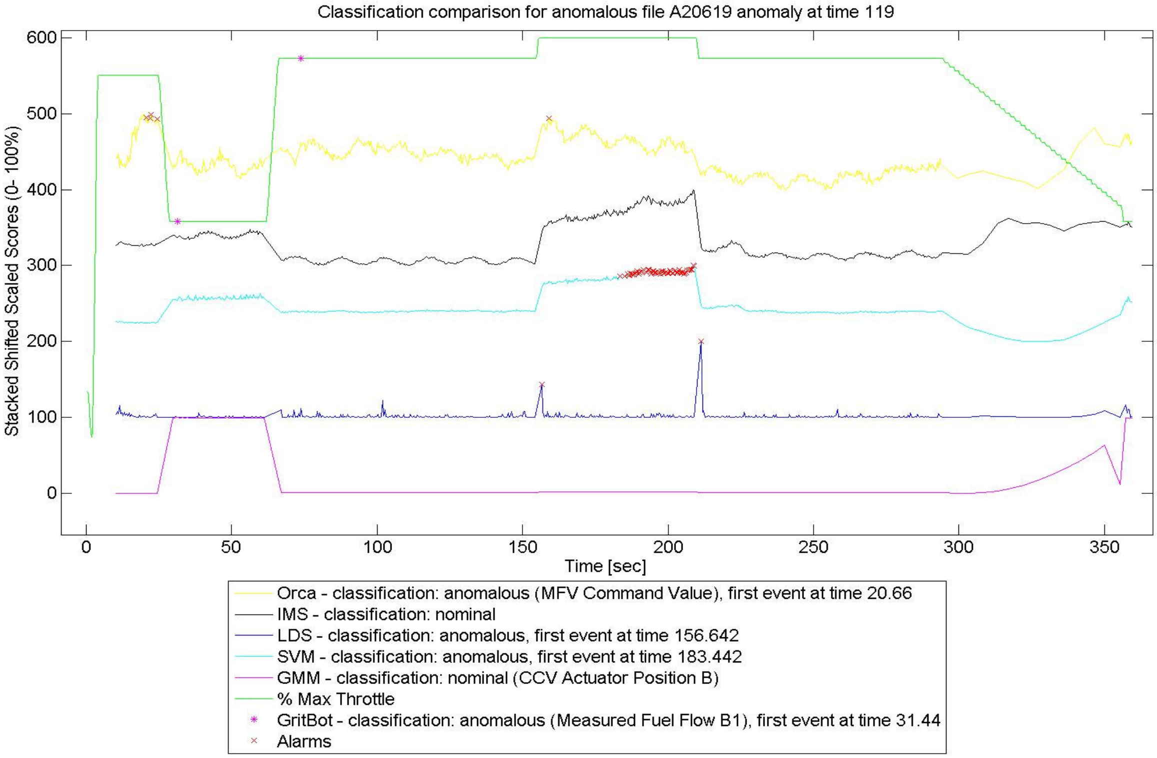


Figure 8: Anomaly Detection Comparison for Validation file A20619

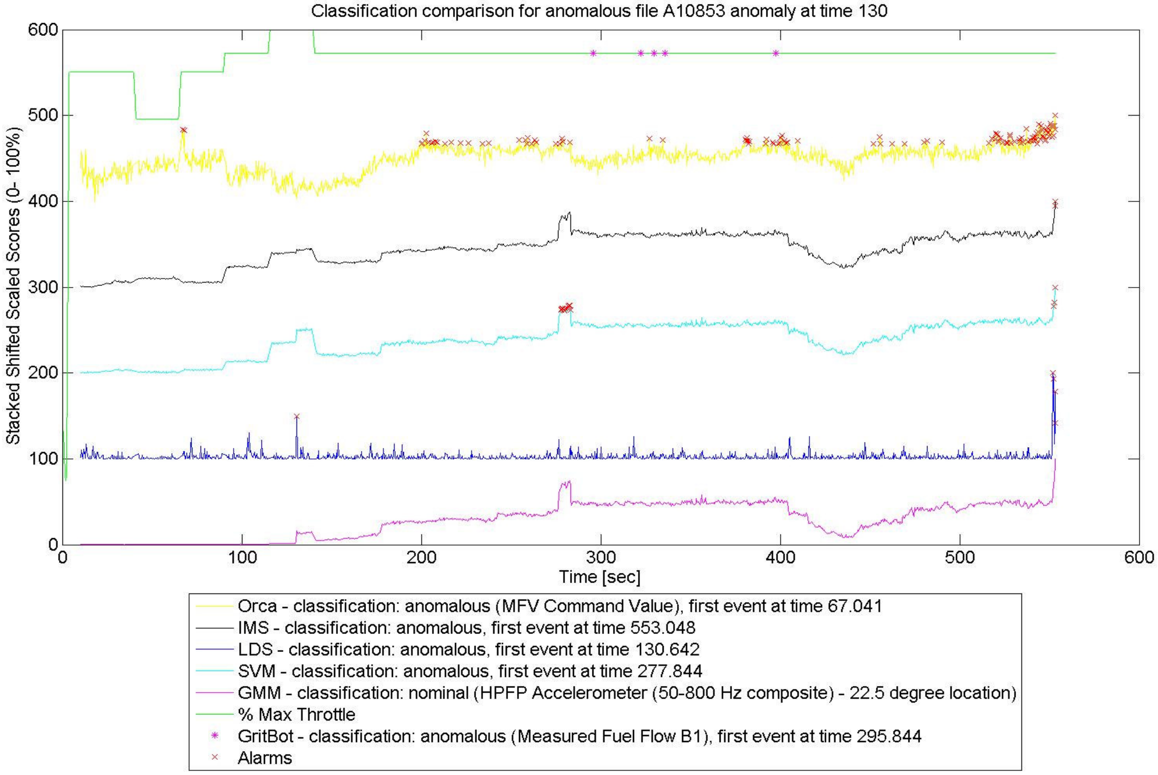


Figure 9: Anomaly Detection Comparison for Validation file A10853

SUMMARY AND CONCLUSIONS

In this paper, we have provided a qualitative discussion pertaining to each of the validation trials presented in Table 2. For the most part, all algorithms have been fairly consistent in their corroboration of the classification of each trial. In only one case was there a tiebreaker required to make any conclusive statement about its final classification, flight STS-106, engine # 2. However, in general we can conclude that by changing the thresholds for various algorithms, their final binary classifications will vary accordingly. There may even potentially be some undiscovered anomalies not highlighted in this paper due to some thresholds being set too high.

As a final comparative summary, we provide confusion matrices and tables on detection times for each algorithm in both the pre-experimental phase and the post-experimental phase after re-classification. In the pre-experimental phase, we use Table 2 as the basis for ground truth to generate the following confusion matrices listed in order of decreasing accuracy and prediction times in Table 3.

Orca: $\begin{bmatrix} 5 & 1 \\ 1 & 5 \end{bmatrix}$, SVM: $\begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix}$, GritBot: $\begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix}$, LDS: $\begin{bmatrix} 3 & 3 \\ 3 & 3 \end{bmatrix}$, IMS: $\begin{bmatrix} 3 & 3 \\ 3 & 3 \end{bmatrix}$, GMM: $\begin{bmatrix} 2 & 4 \\ 4 & 2 \end{bmatrix}$

Failure Data	Failure Type	Time of Failure						
		Actual	Orca	GritBot	SVM	IMS	LDS	GMM
STS-77 (#2)	Anomalous Spike in Sensor Reading	74.42 sec	N/A	N/A	N/A	N/A	N/A	N/A
STS-91 (#1)	Sensor Failure	32.76 sec	77.82 sec	N/A	N/A	N/A	N/A	N/A
STS-93 (#1)	Controller Failure	11.38 sec	25.04 sec	15.44 sec	25.04 sec	25.04 sec	N/A	12.64 sec
STS-93 (#3)	Fuel Leak and Controller Failure	11.62 sec	23.44 sec	176.24 sec	23.44 sec	23.44 sec	17.44 sec	11.84 sec
A20619	Knife Edge Seal Crack	119 sec	20.66 sec	31.44 sec	183.44 sec	N/A	156.64 sec	N/A
A10853	Turbine Blade Failure	130 sec	67.04 sec	295.84 sec	277.84 sec	553.05 sec	130.54 sec	N/A

Table 3: Time of Failures and Detections

The quickest prediction times are highlighted in blue, in Table 3. As an overall assessment, without accounting for the specific circumstances of each particular categorization, it appears that Orca classifies the most accurately, and has quicker prediction times. GritBot and the SVM method are tied for having the second best accuracy, and the GMM method has the second best prediction times for those that are accurately categorized according to the ground truth provided in Table 2. However, in the course of our investigation, we have found that some re-classification of ground truth is necessary in order to best represent the new anomalies identified during analysis. As such, we may add A10852 and STS-103 (#3) to the list of potential anomalies, resulting in the following updated confusion matrices, again listed in order of decreasing accuracy and revised prediction times in Table 4.

SVM: $\begin{bmatrix} 6 & 0 \\ 2 & 4 \end{bmatrix}$, LDS: $\begin{bmatrix} 5 & 1 \\ 3 & 3 \end{bmatrix}$, Orca: $\begin{bmatrix} 5 & 1 \\ 3 & 3 \end{bmatrix}$, IMS: $\begin{bmatrix} 5 & 1 \\ 3 & 3 \end{bmatrix}$, GritBot: $\begin{bmatrix} 5 & 1 \\ 3 & 3 \end{bmatrix}$, GMM: $\begin{bmatrix} 4 & 2 \\ 4 & 2 \end{bmatrix}$

Failure Data	**Failure Type**	**Time of Failure**						
		Actual	**Orca**	**GritBot**	**SVM**	**IMS**	**LDS**	**GMM**
STS-77 (#2)	Anomalous Spike in Sensor Reading	74.42 sec	N/A	N/A	N/A	N/A	N/A	N/A
STS-91 (#1)	Sensor Failure	32.76 sec	77.82 sec	N/A	N/A	N/A	N/A	N/A
STS-93 (#1)	Controller Failure	11.38 sec	25.04 sec	15.44 sec	25.04 sec	25.04 sec	N/A	12.64 sec
STS-93 (#3)	Fuel Leak and Controller Failure	11.62 sec	23.44 sec	176.24 sec	23.44 sec	23.44 sec	17.44 sec	11.84 sec
A20619	Knife Edge Seal Crack	119 sec	20.66 sec	31.44 sec	183.44 sec	N/A	156.64 sec	N/A
A10853	Turbine Blade Failure	130 sec	67.04 sec	295.84 sec	277.84 sec	553.05 sec	130.54 sec	N/A
STS-103 (#3)	Max Noise Failure	38.1, 72.74 sec	N/A	253.84 sec	24.24 sec	24.24 sec	368.65 sec	12.24 sec
A10852	Mixture Ratio Change	210 sec	N/A	N/A	213.025 sec	213.425 sec	210.225 sec	211.425 sec

Table 4: Revised Time of Failures and Detections

Again, the revised quickest prediction times are shown in blue. Now we see that the SVM method appears to have the highest accuracy, followed by all remaining algorithmic techniques tied for second, with the exception of the GMM method. Orca and the GMM method are tied for having the quickest prediction times, followed by the LDS method in second place. These revised results, however, are still subject both to the specific circumstances of each particular categorization and the chosen thresholds.

Independent of the results, all algorithmic methods have both advantages and disadvantages. For example, the profile of the offending sensor for a particular failure is qualitatively present in the score of both the IMS and SVM results. This increases the accuracy of both methods. The GMM method, Orca, and GritBot all have the means to isolate the anomaly to a particular sensor. The LDS method is geared for detecting anomalies in control system error signals, and as such is very sensitive to parameters that influence it (i.e., unexpected mixture ratio changes).

Some disadvantages include the inability of Orca and GritBot to correctly isolate anomalies, specifically when the classifications are correct and early. Another involves the inability of the GMM method to accurately detect anomalies based upon modestly set threshold values, although the same threshold value was used for all sensors. The SVM method and IMS have the ability to detect correctly, but their time to detection is insufficient. Finally, the LDS method cannot isolate sensors, and is best only when detecting anomalies that will present in the control system error.

However, even with all of these disadvantages, there is great potential for development of an architecture and voting logic that leverages all of the advantages of the algorithms. There is a great deal of flexibility in selecting thresholds that best cater to each algorithm. The only reason this wasn't performed here was to provide for a measure of experimental objectivity. Furthermore, real-time implementation of these algorithms would require additional layers of corroboration since the root causes of anomalies are rarely known until a thorough investigation is performed.

FUTURE WORK

Future work that may extend the scope of what was presented here include augmentation of the capabilities of the algorithms that don't have the ability to isolate anomalies to a particular sensor. Of course there are numerous technical endowments that can be implemented, for example, an additional layer of predictive capability can be added to the LDS method as alluded to in previous work.[5] Finally, there is much work that can be performed in order to develop an architecture to support corroboration of potential anomalies, for the ultimate purpose of applying them to future spacecraft propulsion systems.

ACKNOWLEDGMENTS

The authors thank Richard Watson and Bryan Matthews of Perot Systems, Inc. for providing a review and support in development of the tools required to implement several of these algorithms. We also thank John Butas and Anthony Kelly of NASA Marshall Space Flight Center for providing data, and interpretation of some of the results. We thank Matt Davidson, Al Daumann, and John Stephens of Pratt & Whitney Rocketdyne for providing data, as well as Randall Bickford and Edwina Liu of Expert Microsystems for providing data and accompanying detailed descriptions and interpretations of anomalies. The research was supported by the Liquid Propulsion portion of NASA's Integrated System Health Management project within the Exploration Technology Development Program funded by the Exploration Systems Mission Directorate.

REFERENCES

1. H. Park, R. Mackey, M. James, M. Zak, M. Zynard, J. Sebghati, and W. Greene. "Analysis of Space Shuttle Main Engine data using Beacon-based Exception Analysis for Multi-missions," In *Proceedings of the IEEE Aerospace Conference*, Big Sky, MO, March 2002.
2. Stephen D. Bay and Mark Schwabacher. "Mining distance-based outliers in near linear time with randomization and a simple pruning rule," In *KDD '03: Proceedings of The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 29–38, New York, NY, 2003. ACM Press
3. Mark Schwabacher. "Machine learning for rocket propulsion health monitoring," In *Proceedings of the SAE World Aerospace Congress*, volume 114-1, pages 1192–1197, Dallas, Texas, 2005. Society of Automotive Engineers.
4. David L. Iverson. "Inductive system health monitoring," In *Proceedings of The 2004 International Conference on Artificial Intelligence (IC-AI04)*, Las Vegas, Nevada, June 2004. CSREA Press.
5. Rodney A. Martin, "Unsupervised anomaly detection and diagnosis for liquid rocket engine propulsion," In *Proceedings of the IEEE Aerospace Conference*, Big Sky, MO, March 2007.
6. Santanu Das, Ashok Srivastava, and Aditi Chattopadhyah. "Classification of Damage Signatures in Composite Plates using One-Class SVM's," In *Proceedings of the IEEE Aerospace Conference*, Big Sky, MO, March 2007.
7. Gilles Cohen, Melanie Hilario, and Christian Pellegrini. "One-class support vector machines with a conformal kernel. a case study in handling class imbalance," In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 850–858, 2004.
8. Randall Bickford. MSET Signal Validation System Final Report. Technical report, NASA Contract NAS8-98027, August 2000.

9. Adrian Agogino and Kagan Tumer. “Entropy based anomaly detection applied to space shuttle main engines,” In *Proceedings of the IEEE Aerospace Conference*, Big Sky, MO, March 2006.
10. Tony R. Fiorucci, David R. Lakin II, and Tracy D. Reynolds. “Advanced engine health management applications of the SSME real-time vibration monitoring system,” In *Proceedings of the 36th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, Huntsville, AL, July 2000.
11. http://en.wikipedia.org/wiki/STS-93